

Masurare Puls

Autor: ALDEA Lorena Mihaela

Grupa: 334CA

Introducere

Proiectul pe care l-am ales consta in realizarea unui aparat ce masoara pulsul cu diferite functionalitati si anume:

- schimb de culori;
- producere de sunete;

Descriere generală

Pentru realizarea proiectului voi folosi un senzor de puls, un display, un LED tricolor si buzzer.

Odata ce ai pus degetul pe sezor trebuie sa astepti valoarea ce va fi afisata ulterior pe ecran.

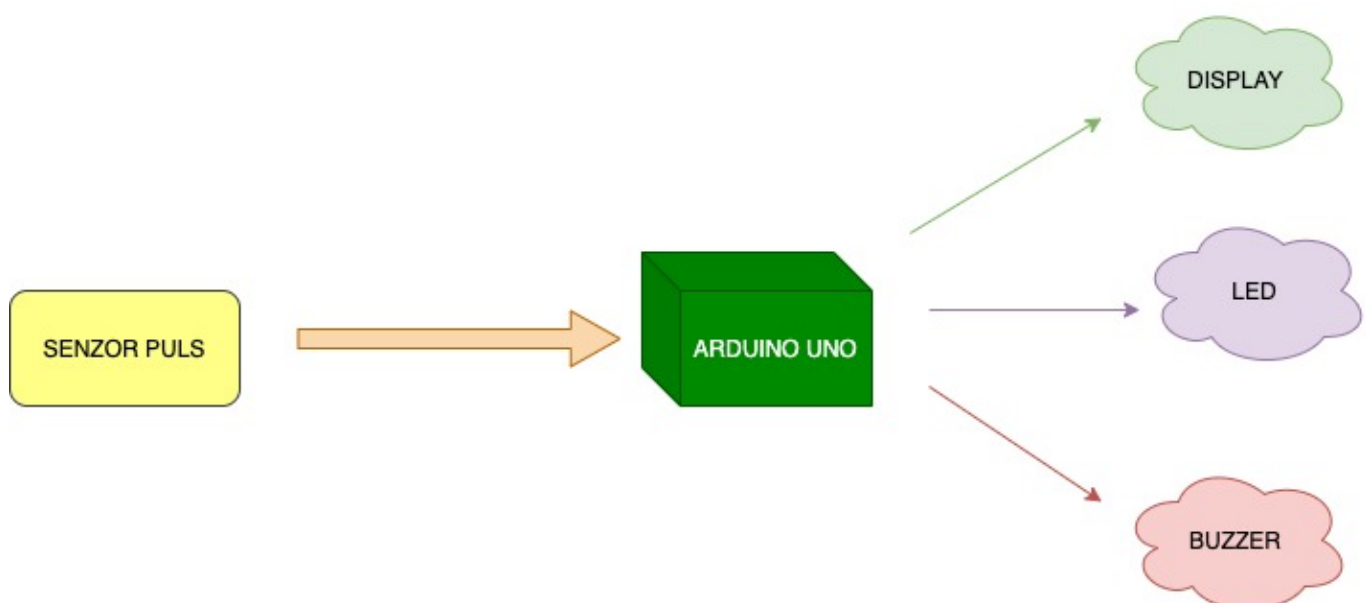
Culoarea LED-ului se va schimba dupa masurarea pulsului astfel incat va fi afisata o culoare in functie de un anumit interval al pulsului:

ROSU, GALBEN SI VERDE !

Hardware Design

Componentele necesare:

- Arduino UNO
- Breadboard
- Buzzer
- LED tricolor
- Senzor puls
- Display OLED
- Fire
- Rezistente de 100 si 220



Pentru a putea realiza partea hardware am urmat urmatoorii pasi:

- Am conectat ARDUINO UNO la laptop

- Am conectat display-ul in functie de indicatiile de pe site (aici am intampinat diferite probleme pe care le-am rezolvat combinand mai multe solutii gasite)
- Am conectat senzorul de puls
- Am atasat si LED-ul tricolor alaturi de cateva rezistente de 220 ohm si fire
- Intr-un final am atasat si buzzerul conform schemei

Software Design

Pentru a putea realiza partea de software am utilizat Arduino IDE.

- Am descarcat toate bibliotecile de care aveam nevoie pentru a face ca proiectul sa functioneze.
- Am implemetat codul pentru display-ul OLED (pentru a putea afisa pe ecran valoarea citita de catre senzor)
- Am initializat senzorul pentru masurarea pulsului la atingere
- Am adaugat si un buzzer ce va scoate un sunet de fiecare data cand culoarea LED-ului este schimbata
- Am conectat LED-ul tricolor ce isi schimba culorile in functie de valorile pe care acesta la masoara astfel:
ROSU - in cazul in care valoarea afisata este > 250
GALBEN - in cazul in care valoarea afisata este > 125
VERDE - in cazul in care valoarea afisata este < 125

```
#define USE_ARDUINO_INTERRUPTS true

#include <Adafruit_GFX.h>          //OLED libraries
#include <Adafruit_SSD1306.h>
#include <Wire.h>
#include <SPI.h>
#include <PulseSensorPlayground.h>

PulseSensorPlayground pulseSensor;
float beatsPerMinute;

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

#define PULSEWIRE 0
#define THRESHOLD 550

#define OLED_DC      9
#define OLED_CS      10
#define OLED_RESET   8

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
    &SPI, OLED_DC, OLED_RESET, OLED_CS);

static const unsigned char PROGMEM logo2_bmp[] =
{ 0x03, 0xC0, 0xF0, 0x06, 0x71, 0x8C, 0x0C, 0x1B, 0x06, 0x18, 0x0E, 0x02,
  0x10, 0x0C, 0x03, 0x10,
  0x04, 0x01, 0x10, 0x04, 0x01, 0x10, 0x40, 0x01, 0x10, 0x40, 0x01, 0x10,
  0xC0, 0x03, 0x08, 0x88,
```

```
0x02, 0x08, 0xB8, 0x04, 0xFF, 0x37, 0x08, 0x01, 0x30, 0x18, 0x01, 0x90,
0x30, 0x00, 0xC0, 0x60,
0x00, 0x60, 0xC0, 0x00, 0x31, 0x80, 0x00, 0x1B, 0x00, 0x00, 0x0E, 0x00,
0x00, 0x04, 0x00,  };

static const unsigned char PROGMEM logo3_bmp[] =
{ 0x01, 0xF0, 0x0F, 0x80, 0x06, 0x1C, 0x38, 0x60, 0x18, 0x06, 0x60, 0x18,
0x10, 0x01, 0x80, 0x08,
0x20, 0x01, 0x80, 0x04, 0x40, 0x00, 0x00, 0x02, 0x40, 0x00, 0x00, 0x02,
0xC0, 0x00, 0x08, 0x03,
0x80, 0x00, 0x08, 0x01, 0x80, 0x00, 0x18, 0x01, 0x80, 0x00, 0x1C, 0x01,
0x80, 0x00, 0x14, 0x00,
0x80, 0x00, 0x14, 0x00, 0x80, 0x00, 0x14, 0x00, 0x40, 0x10, 0x12, 0x00,
0x40, 0x10, 0x12, 0x00,
0x7E, 0x1F, 0x23, 0xFE, 0x03, 0x31, 0xA0, 0x04, 0x01, 0xA0, 0xA0, 0x0C,
0x00, 0xA0, 0xA0, 0x08,
0x00, 0x60, 0xE0, 0x10, 0x00, 0x20, 0x60, 0x20, 0x06, 0x00, 0x40, 0x60,
0x03, 0x00, 0x40, 0xC0,
0x01, 0x80, 0x01, 0x80, 0x00, 0xC0, 0x03, 0x00, 0x00, 0x60, 0x06, 0x00,
0x00, 0x30, 0x0C, 0x00,
0x00, 0x08, 0x10, 0x00, 0x00, 0x06, 0x60, 0x00, 0x00, 0x03, 0xC0, 0x00,
0x00, 0x01, 0x80, 0x00  };

int red_light_pin= 7;
int green_light_pin = 6;
int blue_light_pin = 5;

void setup() {
  Serial.begin(9600);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C); //Start the OLED display
  display.display();
  delay(3000);
  // Initialize sensor
  pulseSensor.analogInput(PULSEWIRE);
  pulseSensor.setThreshold(THRESHOLD);
  pulseSensor.blinkOnPulse(13);

  if (pulseSensor.begin()) {
    Serial.println("We created a pulseSensor Object !");
  }

  pinMode(red_light_pin, OUTPUT);
  pinMode(green_light_pin, OUTPUT);
  pinMode(blue_light_pin, OUTPUT);
}

void RGB_color(int red_light_value, int green_light_value, int
blue_light_value)
{
  analogWrite(red_light_pin, red_light_value);
  analogWrite(green_light_pin, green_light_value);
}
```

```
    analogWrite(blue_light_pin, blue_light_value);
}

void loop() {
    beatsPerMinute = pulseSensor.getBeatsPerMinute();

    display.clearDisplay(); //Clear the
display
    display.drawBitmap(5, 5, logo2_bmp, 24, 21, WHITE); //Draw the
little heart
    display.setTextSize(2); // displays
the average BPM
    display.setTextColor(WHITE);
    display.setCursor(50,0);
    display.println("BPM");
    display.setCursor(50,18);
    display.println(beatsPerMinute);
    display.display();

    RGB_color(255, 255, 255); // White
    delay(1000);

    if(pulseSensor.sawStartOfBeat()){

        display.clearDisplay(); //Clear the
display
        display.drawBitmap(0, 0, logo3_bmp, 32, 32, WHITE); //Draw the
bigger heart
        display.setTextSize(2); // displays the
average BPM
        display.setTextColor(WHITE);
        display.setCursor(50,0);
        display.println("BPM");
        display.setCursor(50,18);
        display.println(beatsPerMinute);
        display.display();
        tone(3,1000);
        delay(100);
        noTone(3); //the "bip" sound

        if (beatsPerMinute > 180) {
            RGB_color(255, 0, 0);
        } else {
            if (beatsPerMinute > 125) {
                RGB_color(255, 255, 0);
            } else {
                RGB_color(0, 255, 0);
            }
        }
    }

    delay(1000);
}
```

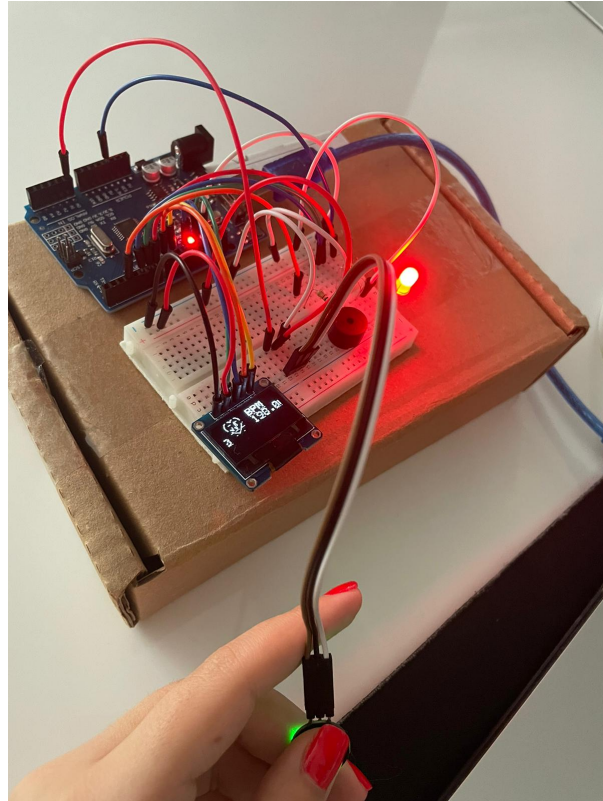
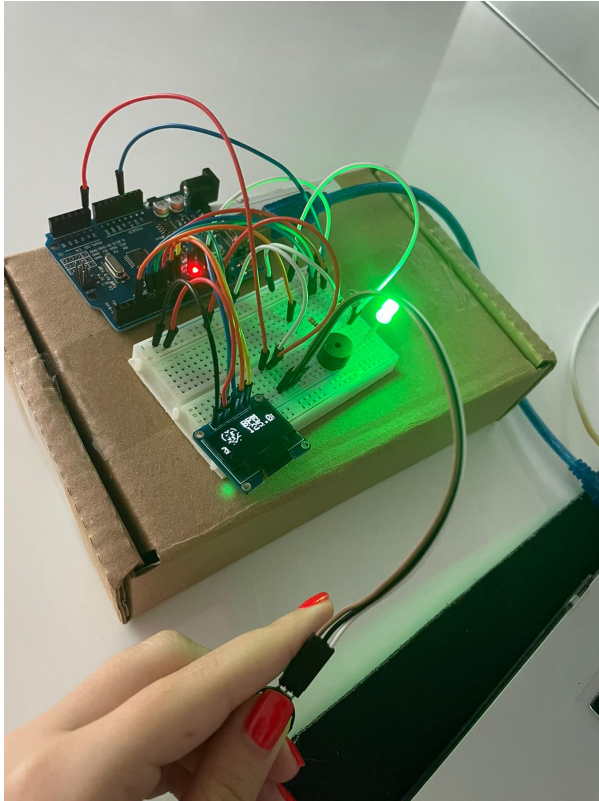
}

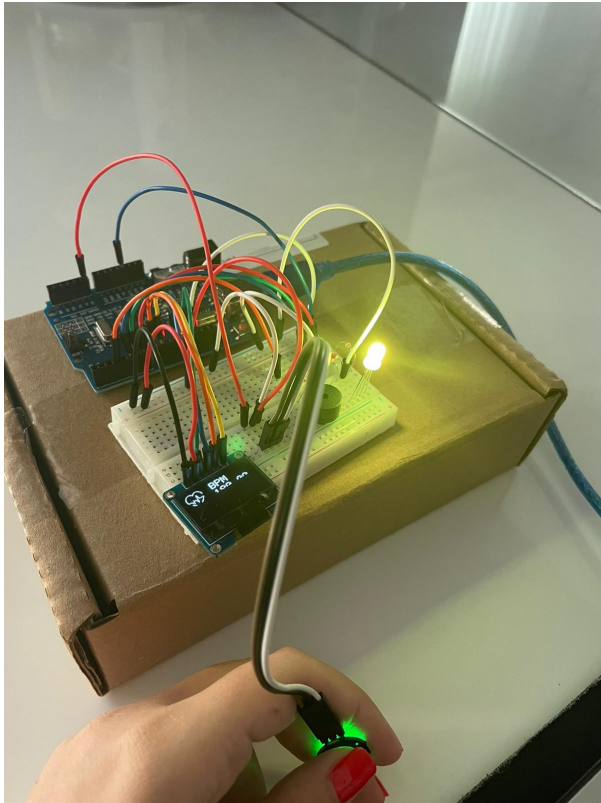
}

Rezultate obtinute

Proiectul face exact ceea ce mi-am propus initial sa realizez.

In micul filmulet pe care l-am realizat se va evidentia cum se schimba culoarea LED-ului in functie de valorile masurate si se va auzi sunetul scos de catre buzzer.





[Demonstratia video a proiectului](#)

Concluzii

A fost o adevarata provocare realizarea acestui proiect, mai ales datorita faptului ca a fost prima data cand m-am atins de partea de hardware, eu nefiind pasionata deloc de aceasta latura.

A fost foarte interesant sa vad cum ceva implementat a inceput "sa prinda viata" in afara computerului meu.

Am invatat sa ma documentez bine inainte de a incepe realizarea proiectului si de asemenea am ramas cu o amintire frumoasa, un proiect interesant din cadrul anului 3 de care cu siguranta imi voi aminti cu placere.

Download

Jurnal

- * 25.04.2021 Alegerea temei proiectului si completarea paginii de wiki cu schema initiala si componentele necesare realizarii proiectului
- * 12.05.2021 -> Comandare pieselor
- * 13.05.2021 -> Sosirea pieselor
- * 20.05.2021 -> Inceperea realizarii proiectului
- * 21.05.2021 -> Finalizarea proiectului
- * 27.05.2021 -> Finalizarea paginii de wiki

Bibliografie/Resurse

[Idea de la care am pornit](#)

[Arduino RGB LED Tutorial](#)

[Utilizare senzor puls](#)

[Buzzer](#)

[masurare_puls.pdf](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/cghenea/masurare-puls>

Last update: **2021/05/29 10:42**

